



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/676,825	09/30/2003	Reiner Hammerich	13913-063001 / 2003P00075	3059
22852	7590	11/13/2006	EXAMINER	
FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER LLP 901 NEW YORK AVENUE, NW WASHINGTON, DC 20001-4413			PHAM, THAI V	
			ART UNIT	PAPER NUMBER
			2191	

DATE MAILED: 11/13/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

**Office Action Summary**

Application No.

10/676,825

Applicant(s)

HAMMERICH ET AL.

Examiner

Thai Van Pham

Art Unit

2192

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 09/30/2003.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1-21 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-21 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 30 September 2003 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☒ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- |   |   |
|---|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)   | 4) <input type="checkbox"/> Interview Summary (PTO-413)<br>Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)  | 5) <input type="checkbox"/> Notice of Informal Patent Application                       |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)<br>Paper No(s)/Mail Date <u>08/16/2004</u> . | 6) <input type="checkbox"/> Other: _____  |

### DETAILED ACTION

This is the initial office action based on the application filed on October 31, 2006.

Priority date that has been considered for this application is September 30, 2003.

Claims 1 – 21 are currently pending and have been considered below.

### ***Claim Rejections - 35 USC § 101***

35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

1. Claims 11 – 21 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

-- Claim 11 recites a computer program product, tangibly embodied in an information carrier as the claimed subject matter. However, The disclosure explicitly states an information carrier can be "*a machine-readable storage device*" or "*a propagated signal*" on page 8, lines 4 – 6.

A product is a tangible physical article or object, some form of matter, which a signal is not. That the other product classes, machine and composition of matter, require physical matter is evidence that a manufacture was also intended to require physical matter. A signal, a form of energy, does not fall within either of the two definitions of manufacture. Thus, a signal does not fall within one of the four statutory classes of U.S.C. 101. (See Interim Guidelines for Examination of Patent Applications for Patent Subject Matter Eligibility (See Interim Guidelines for Examination of Patent Applications for Patent Subject Matter Eligibility (OG Cite: 1300 OG142), Annex IV(c)).

In the principle of compact prosecution, Examiner anticipates the claim will be amended to become statutory claim as such "...A computer program product, tangibly embodies in [an information carrier] a computer-readable storage device..."

Claims 12 – 20: are dependent claims of claim 11. These claims all fail to render the non-statutory claimed subject matter of their parent claim(s) (a computer product) statutory.

***Claim Rejections - 35 USC § 112***

The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

2. Claims 10 and 20 recite the limitation "*generate a second language-dependent program from the language-dependent description*". There is insufficient antecedent basis for this limitation in the claim. Taken the context of the claims and specification into consideration, The Examiner assumes the phrase above is meant to be "*generate a second language-dependent program from the language-independent description*" for further claim analysis under 35 U.S.C. 102 and 103.

Appropriate correction is required.

***Claim Rejections - 35 USC § 102***

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

3. Claims 1 – 21 are rejected under 35 U.S.C. 102(b) as being anticipated by **DeMaster** (US 6,066,181).

-- Claim 1

**DeMaster** discloses *a method for validating programs, the method comprising:*

- *receiving a language-independent description of a computer program, the language-independent description comprising a definition module and an implementation module;*  
(Fig. 1, item 142 and associated text – “IDL file”;

Page 4: lines 18 – 50; “...the programmer initially generates a native interface definition, that preferably uses a neutral Interface Definition Language (IDL) to describe native code components, namely the functions, data structures...”

Page 7: lines 31 – 41; “An IDL specification consists of one or more module definitions, interface definitions, constant definitions or type definitions...”)

- *validating the language-independent description;*

(Fig. 3, page 5: lines 62 – 66; “...the code generator process reads and parses the user-defined native interface definition...”)

- *generating a language-dependent program from the language-independent description, the language-dependent program comprising an interface and a class; and*

(Fig. 1, item 124 and associated text – “Native Code”;

Page 4: lines 51 – 63; “The Java native interface code generator ...generates Java  
Classes and data conversion code stubs...”)

- *validating the language-dependent program.*

(Page 5: lines 55 – 59; “...in order to compile and execute the Java code and native code, the workstation preferably includes the Java Developer’s Kit and a native code compiler...”)

-- Claim 2.

**DeMaster** discloses *the method of claim 1,*

- *wherein validating the language-independent description comprises validating the syntax of the definition module and the implementation module.*

(Page 6: lines 15 – 18; “...the user interface initially reads and parses the user-derived native interface definition ...to generate mappings for everything in the parse tree.”)

-- Claim 3.

**DeMaster** discloses *the method of claim 1,*

- *wherein validating the language-dependent program comprises compiling the interface and the class.*

(Page 7: lines 12 – 16; “...the generated Java classes ...as well as the Java application program is compiled with JDK, while the data conversion code stubs ...are compiled into a Dynamic Linked Library (DLL)...”)

-- Claim 4.

**DeMaster** discloses *the method of claim 1,*

- *wherein the definition module and the implementation module are represented in a meta-language or using a tree structure.*

(Fig. 3, page 4: lines 51 – 63; “The Java native interface code generator ...generates Java Classes and data conversion code stubs...”)

-- Claim 5.

**DeMaster** discloses *a method for validating programs, the method comprising:*

- *receiving a language-independent description of a computer program, the language-independent description comprising a definition module and an implementation module;*

(Fig. 1, item 142 and associated text – “IDL file”;

Page 4: lines 18 – 50; “...the programmer initially generates a native interface definition, that preferably uses a neutral Interface Definition Language (IDL) to describe native code components, namely the functions, data structures...”

Page 7: lines 31 – 41; “An IDL specification consists of one or more module definitions, interface definitions, constant definitions or type definitions...”)

- *validating the language-independent description;*

(Fig. 3, page 5: lines 62 – 66; “...the code generator process reads and parses the user-defined native interface definition...”)

- *generating a language-dependent program from the language-independent description, the language-dependent program comprising a script code section; and*

(Fig. 1, item 124 and associated text – “Native Code”;

Page 4: lines 51 – 63; “The Java native interface code generator ...generates Java Classes and data conversion code stubs...”)

- *validating the language-dependent program.*

(Page 5: lines 55 – 59; “...in order to compile and execute the Java code and native code, the workstation preferably includes the Java Developer’s Kit and a native code compiler...”)

-- Claim 6.

**DeMaster** discloses *the method of claim 5, wherein validating the language-dependent program comprises:*

- *extracting language elements from the script code section; and*
- *comparing the extracted language elements with the definition module.*

(Fig. 3, page 5: lines 9 – 16; “The data conversion code stubs (JNI code) convert and maps the native data and functions between the native language and Java.” Page 7: lines 12 – 16; “...the data conversion code stubs (JNI functions) ...are compiled into a Dynamic Linked Library (DLL)...”)

-- Claim 7.

**DeMaster** discloses *the method of claim 6,*

- *wherein extracting language elements comprises generating a symbol table from the script code section.*

(When the data conversion code stubs are compiled into a library, symbol table must inherently be built from the code during compilation.)



-- Claim 8.

**DeMaster** discloses *the method of claim 5,*

- *wherein generating the language-dependent program comprises generating language-dependent code comprising an interface and a class.*

(Page 7: lines 12 – 16; "...the generated Java classes ...as well as the Java application program is compiled with JDK, while the data conversion code stubs ...are compiled into a Dynamic Linked Library (DLL)...")

-- Claim 9.

**DeMaster** discloses *the method of claim 5, wherein validating the language-dependent program comprises:*

- *extracting language elements from the script code section;*
- *comparing the extracted language elements with the definition module;*

(Fig. 3, page 5: lines 9 – 16; "The data conversion code stubs (JNI code) convert and maps the native data and functions between the native language and Java." Page 7: lines 12 – 16; "...the data conversion code stubs (JNI functions) ...are compiled into a Dynamic Linked Library (DLL)...")

- *generating language-dependent code comprising an interface and a class; and*

(Page 7: lines 12 – 16; "...the generated Java classes ...as well as the Java application program is compiled with JDK, while the data conversion code stubs ...are compiled into a Dynamic Linked Library (DLL)...")

- *compiling the interface and the class.*

(Page 7: lines 12 – 16; "...the generated Java classes ...as well as the Java application program is compiled with JDK, while the data conversion code stubs ...are compiled into a Dynamic Linked Library (DLL)...")

-- Claim 10.

**DeMaster** discloses *a method for validating programs, the method comprising:*

- *receiving a language-independent description of a computer program, the language-independent description comprising a definition module and an implementation module;*

(Fig. 1, item 142 and associated text – "IDL file";

Page 4: lines 18 – 50; "...the programmer initially generates a native interface definition, that preferably uses a neutral Interface Definition Language (IDL) to describe native code components, namely the functions, data structures..."

Page 7: lines 31 – 41; "An IDL specification consists of one or more module definitions, interface definitions, constant definitions or type definitions...")

- *validating the language-independent description;*

(Fig. 3, page 5: lines 62 – 66; "...the code generator process reads and parses the user-defined native interface definition...")

- *generating a first language-dependent program from the language-independent description, the first language-dependent program comprising a first script code section;*

(Fig. 1, item 124 and associated text – "Native Code";

Page 4: lines 51 – 63; "The Java native interface code generator ...generates Java Classes and data conversion code stubs...")

- *generating a second language-dependent program from the language-dependent description, the second language-dependent program comprising a second script code section of a distinct, second kind;*

(Fig. 1 item 128 – “Data Conversion Code Stubs”;

Page 5: lines 10 – 16; “The data conversion code stubs (JNI code) convert and maps the native data and functions between the native language and Java.” Page 2: lines 5 – 10; “...native code programmed in a native language, such as C, C++ or Assembly....”)

- *extracting a first set of language elements from the first script code section;*

(Fig. 10B and associated text; “The Java class for JmuddStructTest”).)

- *extracting a second set of language elements from the second script code section;*

and

(Fig. 10C and associated text; “The data conversion code stubs for the JmuddStructTest class.)

- *comparing the first set of language elements and the second set of language elements with the definition module.*

(Fig. 3, page 5: lines 9 – 16; “The data conversion code stubs (JNI code) convert and maps the native data and functions between the native language and Java.” Page 7: lines 12 – 16; “...the data conversion code stubs (JNI functions) ...are compiled into a Dynamic Linked Library (DLL)...”)

-- Claims 11 – 19: are computer product claims for performing a method corresponding to the method of claims 1 – 9, respectively; Therefore, claims 11 – 19 are rejected for the same reason set forth in connection to the rejection of claims 1 – 9 above, respectively.

-- Claim 20: is a computer product claim for performing a method corresponding to the method of claim 10. Therefore, claim 20 is rejected for the same reason set forth in connection to the rejection of claim 10 above.

-- Claim 21:

**Examiner's Note:** it appears that the Applicant is attempting to invoke 35 U.S.C. 112, 6<sup>th</sup> paragraph, with the use of means-plus-function language in claim 21. The specification does not provide any specific or special physical structure(s) for the features that could be read into the claim to limit the scope of the means to perform the claimed functions. The specification, however, discloses that all the "means for" receiving, generating, and validating can be performed by "a general or special purpose microprocessor" or "special logic circuitry, e.g., an FPGA, or an ASIC" (page 8: line 15 – page 9: line5). Therefore, Examiner considers the system of claim 21 to be a computer system containing a microprocessor and/or FPGA/ASIC for performing the method similar to that of claim 1.

**DeMaster** discloses an apparatus (Fig. 2, item 200 and associated text – "developer's workstation") for performing a method corresponding to the method of claim 1.

**Conclusion**

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Thai Van Pham whose telephone number is (571) 270-1064. The examiner can normally be reached on Monday - Thursday, 9am - 5pm EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.



TUAN DAM  
SUPERVISORY PATENT EXAMINER

TVP  
October 31, 2006